

Figure 1

13 22 24 26 20 32

RemoteCogCenter [funmethods.c]

```
*****  
***** COGNIZANT DESIGN GROUP INC. *****  
This file contains user methods.  
*****  
***** Include Files *****  
***** Define *****  
***** Internal Global Variables *****  
/* Set variable to 1 if application is thread safe */  
#ifdef __cplusplus  
extern "C" int bRCogThreadSafe;  
#endif  
int bRCogThreadSafe = 1;  
***** External Variables *****  
***** Forward Declares *****  
***** METHOD LIST DECLARATIONS *****  
*****
```

18 28 30

Figure 2

卷之三

26

43

The screenshot shows a software application window with the following components:

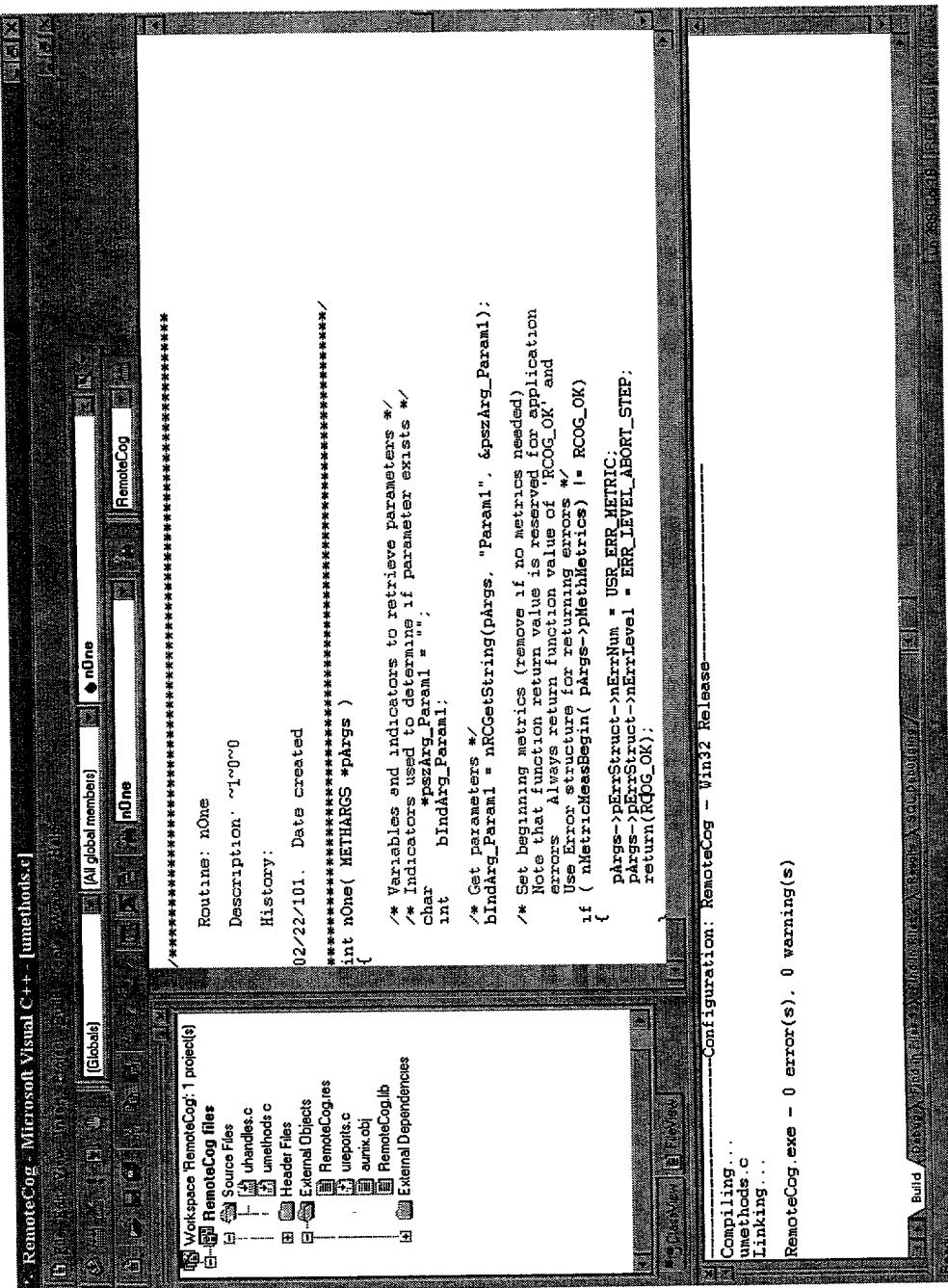
- Top Bar:** Includes a 'File' menu with 'Exit' and 'Help' options, and a 'Edit' menu with 'Copy', 'Paste', 'Delete', 'Select All', and 'Find'.
- Tree View (Left):** A hierarchical tree structure showing scenarios and their sub-components. The 'Demo' scenario is expanded, showing 'FailureSimulation', 'Mail', 'ODBC Toolkit', 'PasteDemo', 'SET NUM CLIENT', 'STOP DAEMON', 'CONFIG', 'One', 'Four', 'Two', and 'Three'.
- Method Parameters (Center):** A table with the following rows:

Method Parameters	Value
Unit	
nOne	
Four	
Two	
Three	
Param1	
- Scenario tasks (Bottom):** A list of tasks:
 - RemoteCG
 - Scenario tasks
 - Capture Activity
 - Prepare Scenario
 - Run Scenario
 - Other tasks
 - System Monitor
 - Configuration
 - Publisher
 - Real Time Info
 - Transaction Stat
 - Graph Trans Rpt
- Bottom Right:** A 'Log' button.

42

Figure 3

Figure 4



The screenshot shows the Microsoft Visual Studio IDE interface. On the left, the Solution Explorer displays the 'RemoteCog' project, which contains one solution and one project named 'RemoteCog'. The 'RemoteCog' project is expanded, showing 'Source Files' (containing 'unmethods.c'), 'Header Files' (containing 'uhandles.c'), 'External Objects' (containing 'RemoteCog.res'), and 'External Dependencies' (containing 'aumk.obj' and 'RemoteCog.lib'). The 'Properties' tab is selected for the project. On the right, the Code Editor displays the 'lunmethods.c' file. The code is a C program with the following structure:

```
*****  
Routine: nOne  
Description: ~1^~0~0  
History: 02/22/111. Date created  
*****  
int nOne( METHARGS *pArgs )  
{  
    /* Variables and indicators to retrieve parameters */  
    /* Indicators used to determine if parameter exists */  
    char *pszArg_Parml = "";  
    int bIndArg_Parml;  
  
    /* Get parameters */  
    bIndArg_Parml = nRGGetString(pArgs, "Param1", &pszArg_Parml);  
  
    /* Set beginning metrics (remove if no metrics needed)  
     * Note that function return value is reserved for application  
     * errors. Always return function value of 'RCOG_OK' and  
     * use error structure for returning errors */  
    if ( nMetricsBegin( pArgs->pMetrics ) != RCOG_OK )  
    {  
        pArgs->ErrStruct->nErrNum = USR_ERR_METRIC;  
        pArgs->ErrStruct->nErrLevel = ERR_LEVEL_ABORT_STEP;  
        return( RCGG_OK );  
    }  
}  
-----Configuration: RemoteCog - Win32 Release-----  
Compiling...  
unmethods.c  
Linking...  
RemoteCog.exe - 0 error(s). 0 warning(s)
```

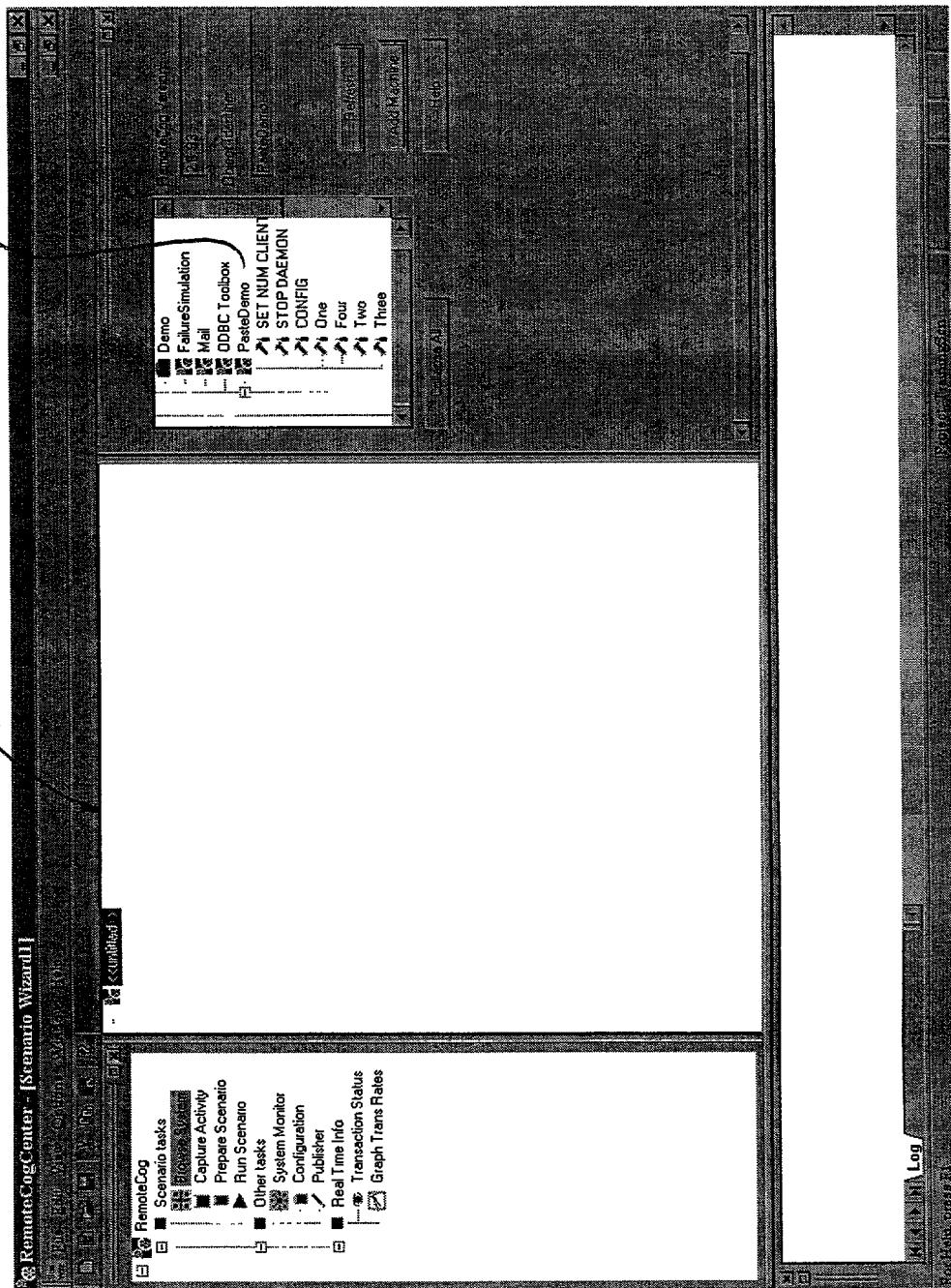
44
28

Figure 5

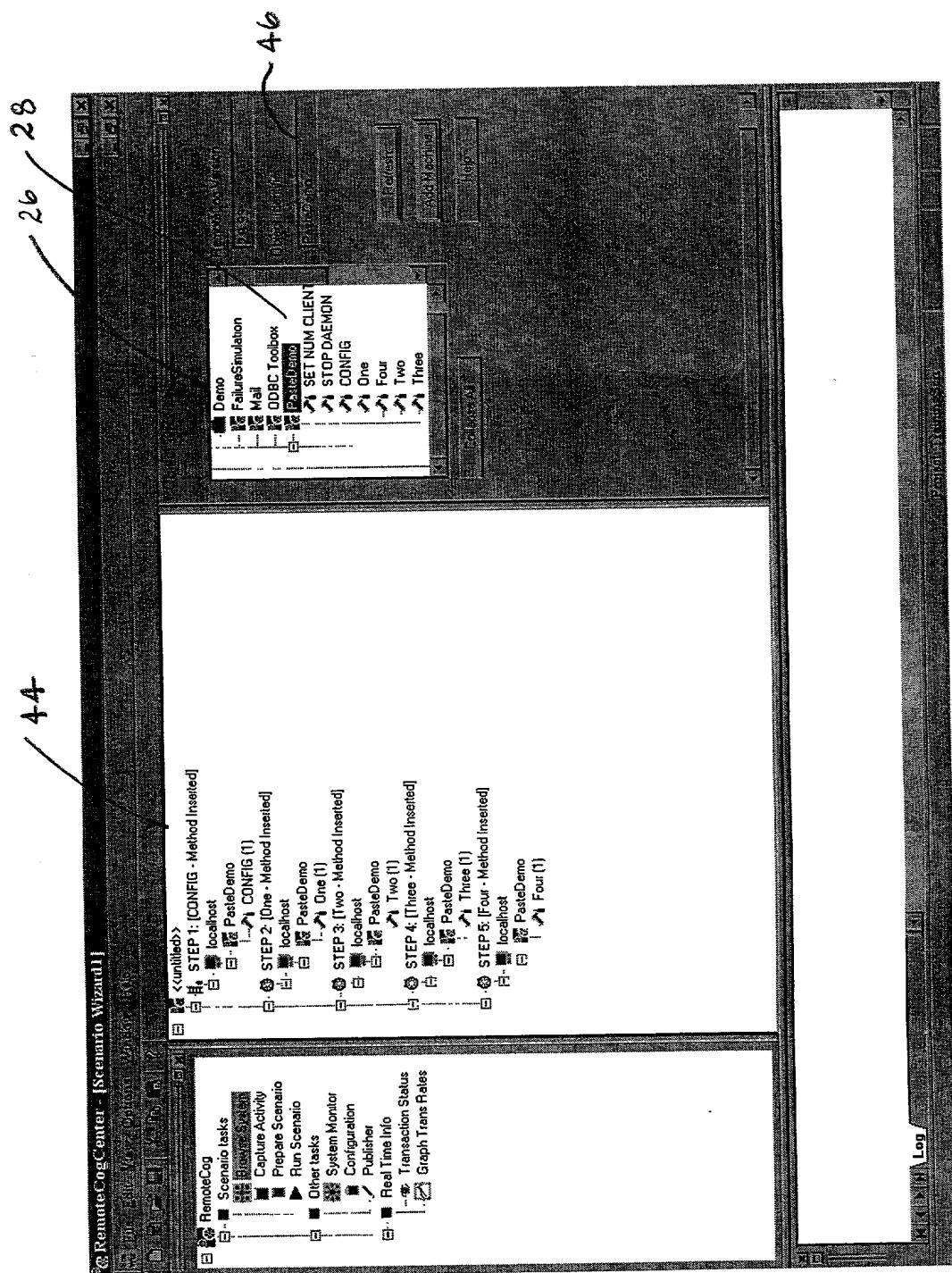


Figure 6

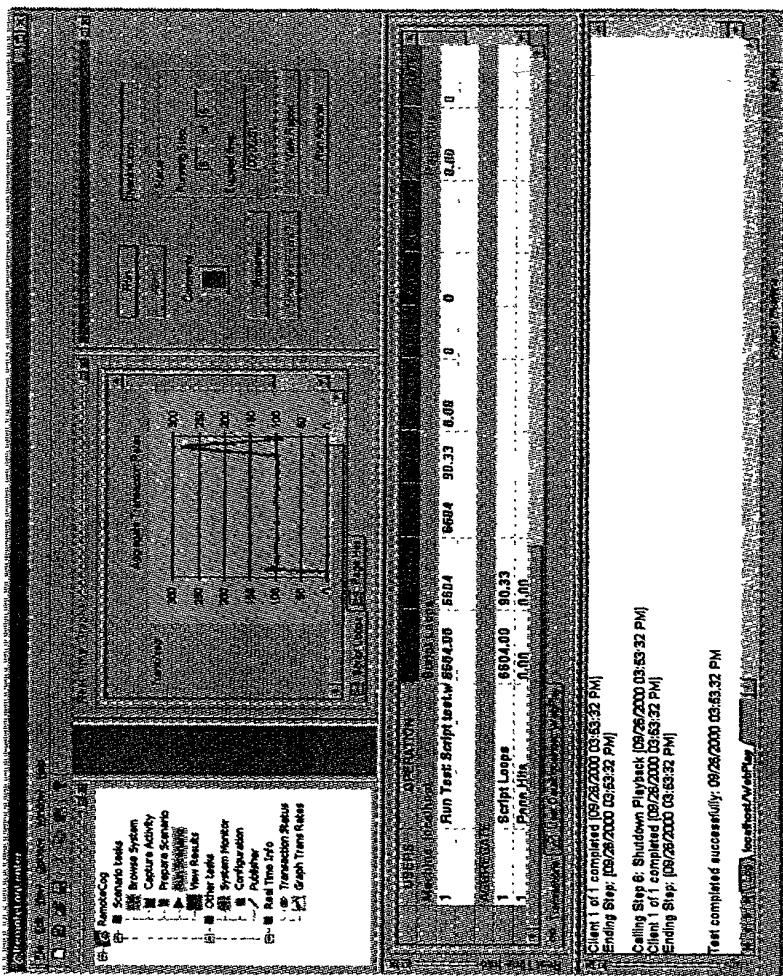


Figure 7